JASPERREPORTS - CREATE SUBREPORTS

http://www.tutorialspoint.com/jasper_reports/jasper_create_subreports.htm

Subreports are one of the nice features of the JasperReports. This feature allows incorporating a report within another report, that is, one report can be a subreport of another. Subreports help us keep report designs simple, as we can create many simple reports and encapsulate them into a master report. Subreports are compiled and filled just like normal reports. Any report template can be used as a subreport when incorporated into another report template, without anything inside it having to change.

Subreports are like normal report templates. They are in fact *net.sf.jasperreports.engine.JasperReport* objects, which are obtained after compiling a *net.sf.jasperreports.engine.design.JasperDesign object*.

<subreport> Element

A <subreport> element is used when introducing subreports into master reports. Here is the list of sub-elements in the <subreport> JRXML element.

- <reportElement>
- <parametersMapExpression> : This is used to pass a map containing report parameters to the subreport. The map is usually obtained from a parameter in the master report, or by using the built-in REPORTS_PARAMETERS_MAP parameter to pass the parent report's parameters to the subreport. This expression should always return a *java.util.Map* object in which the keys are the parameter names.
- <subreportParameter> : This element is used to pass parameters to the subreport. It has an attribute *name*, which is mandatory.
- <connectionExpression > : This is used to pass a *java.sql.Connection* to the subreport. It is used only when the subreport template needs a database connection during report filling phase.
- <dataSourceExpression> : This is used to pass a datasource to the subreport. This datasource is usually obtained from a parameter in the master report or by using the built-in REPORT_DATA_SOURCE parameter to pass the parent report's datasource to the subreport.

  The elements (*connectionExpression and dataSourceExpression*) cannot be present at the same time in a <subreport> element declaration. This is because we cannot supply both a data source and a connection to the subreport. We must decide on one of them and stick to it.
- <returnValue> : This is used to assign the value of one of the subreport's variables to one of the master report's variables. This sub element has attributes as follows:
  - *subreportVariable*: This attribute specifies the name of the subreport variable whose value is to be returned.
  - *toVariable*: This attribute specifies the name of the parent report variable whose value is to be copied/incremented with the value from the subreport.
  - *calculation*: This attribute can take values : Nothing, Count, DistinctCount, Sum, Average, Lowest, Highest, StandardDeviation, Variance. Default value for attribute *calculation* is "Nothing".
  - *incrementerFactoryClass*: This attribute specifies the factory class for creating the incrementer instance.
- <subreportExpression> : This indicates where to find the compiled report template for the subreport. This element has a **class** attribute. The *class* attribute can take any of these values:java.lang.String, java.io.File,

java.net.URL, java.io.InputStream, net.sf.jasperreports.engine.JasperReport.Default value is *java.lang.String*.

- isUsingCache : This is an attribute of the <subreport> element. This is a Boolean, when set to *true*, the reporting engine will try to recognize previously loaded subreport template objects, using their specified source. This caching functionality is available only for subreport elements that have expressions returning java.lang.String objects as the subreport template source, representing file names, URLs, or classpath resources.

Example

Let take up a simple example to demonstrate creation of subreports using JRDataSource. Let's first write two new report templates, one being subreport and the other Master report. The contents of subreport (address_report_template.jrxml) template is as below. Save it to C:\tools\jasperreports-5.0.1\test directory.

```
<?xml version="1.0" encoding="UTF-8"?>
<jasperReport
xmlns="http://jasperreports.sourceforge.net/jasperreports"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://jasperreports.sourceforge.net/jasperreports
http://jasperreports.sourceforge.net/xsd/jasperreport.xsd"
name="address_report_template" pageWidth="175" pageHeight="842" columnWidth="175"
leftMargin="0" rightMargin="0" topMargin="0" bottomMargin="0">
   <field name="city" />
   <field name="street" />
   <background>
         <band splitType="Stretch"/>
   </background>
   <title>
      <band height="20" splitType="Stretch">
         <staticText>
            <reportElement x="0" y="0" width="100" height="20"/>
            <textElement>
               <font size="14" isBold="true"/>
            </textElement>
            <text><![CDATA[Addresses]]></text>
         </staticText>
      </band>
   </title>
   <pageHeader>
      <band height="12" splitType="Stretch"/>
   </pageHeader>
   <columnHeader>
      <band height="12" splitType="Stretch"/>
   </columnHeader>
   <detail>
      <band height="27" splitType="Stretch">
         <textField>
            <reportElement x="0" y="0" width="120" height="20"/>
            <textElement>
               <font size="12" isBold="true"/>
            </textElement>
            <textFieldExpression >
               <![CDATA[$F{city}+" Address:"]]>
            </textFieldExpression>
         </textField>
         <textField isStretchWithOverflow="true">
            <reportElement x="120" y="0" width="435" height="20"/>
            <textElement>
               <font size="12"/>
            </textElement>
            <textFieldExpression >
               <![CDATA[$F{street}]]>
            </textFieldExpression>
         </textField>
      </band>
   </detail>
   <columnFooter>
      <band height="8" splitType="Stretch"/>
   </columnFooter>
   <pageFooter>
      <band height="11" splitType="Stretch"/>
   </pageFooter>
   <summary>
      <band height="9" splitType="Stretch"/>
   </summary>

</jasperReport>
```

As we use a data source we need to write a corresponding POJO file **SubReportBean.java** as below. Save it to directory C:\tools\jasperreports-5.0.1\test\src\com\tutorialspoint:

```
package com.tutorialspoint;

public class SubReportBean {
   private String city;
   private String street;

   public String getCity() {
      return city;
   }

   public void setCity(String city) {
      this.city = city;
   }

   public String getStreet() {
      return street;
   }

   public void setStreet(String street) {
      this.street = street;
   }
}
```

Here we have declared two fields' *city* and *street* and respective getter and setter methods are defined.

Now let's update our existing **DataBean** file. We will add a new field *subReportBeanList* which is a java.util.List. This field will hold the list of SubReportBean objects. The contents of the file DataBean are as below. Save it to directory C:\tools\jasperreports-5.0.1\test\src\com\tutorialspoint.

```
package com.tutorialspoint;

import java.util.List;

public class DataBean {
   private String name;
   private String country;
   private List<SubReportBean> subReportBeanList;

   public String getName() {
      return name;
   }

   public void setName(String name) {
      this.name = name;
   }

   public String getCountry() {
      return country;
   }

   public void setCountry(String country) {
      this.country = country;
   }

   public List<SubReportBean> getSubReportBeanList() {
      return subReportBeanList;
   }

   public void setSubReportBeanList(List<SubReportBean> subReportBeanList) {
      this.subReportBeanList = subReportBeanList;
   }
}
```

Let's now update the file C:\tools\jasperreports-5.0.1\test\src\com\tutorialspoint\DataBeanList.java. The contents of this file are as below:

```
package com.tutorialspoint;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class DataBeanList {
   public ArrayList<DataBean> getDataBeanList() {

      // Create sub report data
      SubReportBean subBean1 = new SubReportBean();
      subBean1.setCity("Mumbai");
      subBean1.setStreet("M.G.Road");
      SubReportBean subBean2 = new SubReportBean();
      subBean2.setCity("New York");
      subBean2.setStreet("Park Street");
      SubReportBean subBean3 = new SubReportBean();
      subBean3.setCity("San Fransisco");
      subBean3.setStreet("King Street");

      ArrayList<DataBean> dataBeanList =
            new ArrayList<DataBean>();

      // Create master report data
      dataBeanList.add(produce("Manisha", "India",
            Arrays.asList(subBean1)));
      dataBeanList.add(produce("Dennis Ritchie", "USA",
            Arrays.asList(subBean2)));
      dataBeanList.add(produce("V.Anand", "India",
            Arrays.asList(subBean1)));
      dataBeanList.add(produce("Shrinath", "California",
            Arrays.asList(subBean3)));

      return dataBeanList;
   }

   /*
    * This method returns a DataBean object,
    * with name, country and sub report
    * bean data set in it.
    */
   private DataBean produce(String name, String country,
         List<SubReportBean> subBean) {
      DataBean dataBean = new DataBean();

      dataBean.setName(name);
      dataBean.setCountry(country);
      dataBean.setSubReportBeanList(subBean);

      return dataBean;
   }
}
```

In the method produce() in the above file, we are setting the list of SubReportBean.

Now let's write a new master report template (jasper_report_template.jrxml). Save this file to directory **C:\tools\jasperreports-5.0.1\test**. The contents for this file are as below:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<jasperReport xmlns="http://jasperreports.sourceforge.net/jasperreports"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://jasperreports.sourceforge.net/jasperreports
http://jasperreports.sourceforge.net/xsd/jasperreport.xsd"
name="jasper_report_template" language="groovy" pageWidth="595"
pageHeight="842" columnWidth="555" leftMargin="20" rightMargin="20"
topMargin="20" bottomMargin="20">

   <parameter name="SUBREPORT_DIR"
      isForPrompting="false">
      <defaultValueExpression>
      <![CDATA["C:\\tools\\jasperreports-5.0.1\\test\\"]]>
      </defaultValueExpression>
   </parameter>
   <field name="country" />
   <field name="name" />
   <field name="subReportBeanList" />
   <background>
      <band splitType="Stretch"/>
   </background>
   <title>
      <band height="35" splitType="Stretch">
         <staticText>
            <reportElement x="0" y="0" width="204" height="34"/>
            <textElement>
            <font size="26" isBold="true"/>
            </textElement>
            <text><![CDATA[Contact Report]]></text>
         </staticText>
      </band>
   </title>
   <pageHeader>
      <band height="17" splitType="Stretch"/>
   </pageHeader>
   <columnHeader>
      <band height="21" splitType="Stretch"/>
   </columnHeader>
   <detail>
      <band height="112" splitType="Stretch">
            <staticText>
            <reportElement x="0" y="0" width="100" height="20"/>
            <textElement>
               <font size="12" isBold="true"/>
            </textElement>
            <text><![CDATA[Name:]]></text>
         </staticText>
         <staticText>
            <reportElement x="0" y="20" width="100" height="20"/>
            <textElement>
               <font size="12" isBold="true"/>
            </textElement>
            <text><![CDATA[Country:]]></text>
         </staticText>
         <textField>
         <reportElement x="104" y="0" width="277" height="20"/>
            <textElement>
               <font size="12"/>
            </textElement>
            <textFieldExpression >
               <![CDATA[$F{name}]]>
            </textFieldExpression>
         </textField>
         <textField>
            <reportElement x="104" y="20" width="277" height="20"/>
            <textElement>
            <font size="12"/>
            </textElement>
            <textFieldExpression >
               <![CDATA[$F{country}]]>
            </textFieldExpression>
         </textField>
         <subreport>
            <reportElement positionType="Float" x="335" y="25" width="175"
            height="20" isRemoveLineWhenBlank="true"
            backcolor="#99ccff"/>

            <dataSourceExpression>
            new net.sf.jasperreports.engine.data.JRBeanCollectionDataSource
            ($F{subReportBeanList})
            </dataSourceExpression>
            <subreportExpression >
               <![CDATA[$P{SUBREPORT_DIR} + "address_report_template.jasper"]]>
            </subreportExpression>
         </subreport>
         <line>
            <reportElement x="0" y="50" width="550" height="1"/>
         </line>
      </band>
   </detail>
   <columnFooter>
      <band height="19" splitType="Stretch"/>
   </columnFooter>
   <pageFooter>
      <band height="18" splitType="Stretch"/>
   </pageFooter>
   <summary>
      <band height="14" splitType="Stretch"/>
   </summary>
</jasperReport>
```

In the above template, we have defined a new parameter "SUBREPORT_DIR" which defines the path of the subreport. We have defined a field *subReportBeanList* of type *java.util.List* which corresponds to property in the file DataBean. The element <subreport> has sub-element <dataSourceExpression>. We have put the list *subReportBeanList* in an instance of JRBeanCollectionDataSource. In the sub-element <subreportExpression/>, we have give the subreport name (AddressReport.jasper).

Now let's write a new class **CreateReport** to compile and execute our report template. The contents of file **C:\tools\jasperreports-5.0.1\test\src\com\tutorialspoint\CreateReport.java** are as below:

```
package com.tutorialspoint;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

import net.sf.jasperreports.engine.JRException;
import net.sf.jasperreports.engine.JasperCompileManager;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.engine.data.JRBeanCollectionDataSource;

public class CreateReport {

   public static void main(String[] args) {
      String masterReportFileName = "C://tools/jasperreports-5.0.1/test"
            + "/jasper_report_template.jrxml";
      String subReportFileName = "C://tools/jasperreports-5.0.1/test"
            + "/AddressReport.jrxml";
      String destFileName = "C://tools/jasperreports-5.0.1/test"
            + "/jasper_report_template.JRprint";
      DataBeanList DataBeanList = new DataBeanList();
      ArrayList<DataBean> dataList = DataBeanList.getDataBeanList();
      JRBeanCollectionDataSource beanColDataSource =
            new JRBeanCollectionDataSource(dataList);

      try {
         /* Compile the master and sub report */
         JasperReport jasperMasterReport = JasperCompileManager
             .compileReport(masterReportFileName);
         JasperReport jasperSubReport = JasperCompileManager
             .compileReport(subReportFileName);
         Map<String, Object> parameters = new HashMap<String, Object>();
         parameters.put("subreportParameter", jasperSubReport);
         JasperFillManager.fillReportToFile(jasperMasterReport,
            destFileName, parameters, beanColDataSource);

      } catch (JRException e) {

         e.printStackTrace();
      }
      System.out.println("Done filling!!! ...");
   }
}
```

Here we see that we are compiling both the master and sub report templates and passing the master report (.jasper) file for report filling.

Report Generation

Now that all our files are ready, let's compile and execute them using our regular ANT build process. The contents of the file build.xml (saved under directory C:\tools\jasperreports-5.0.1\test) are as below.

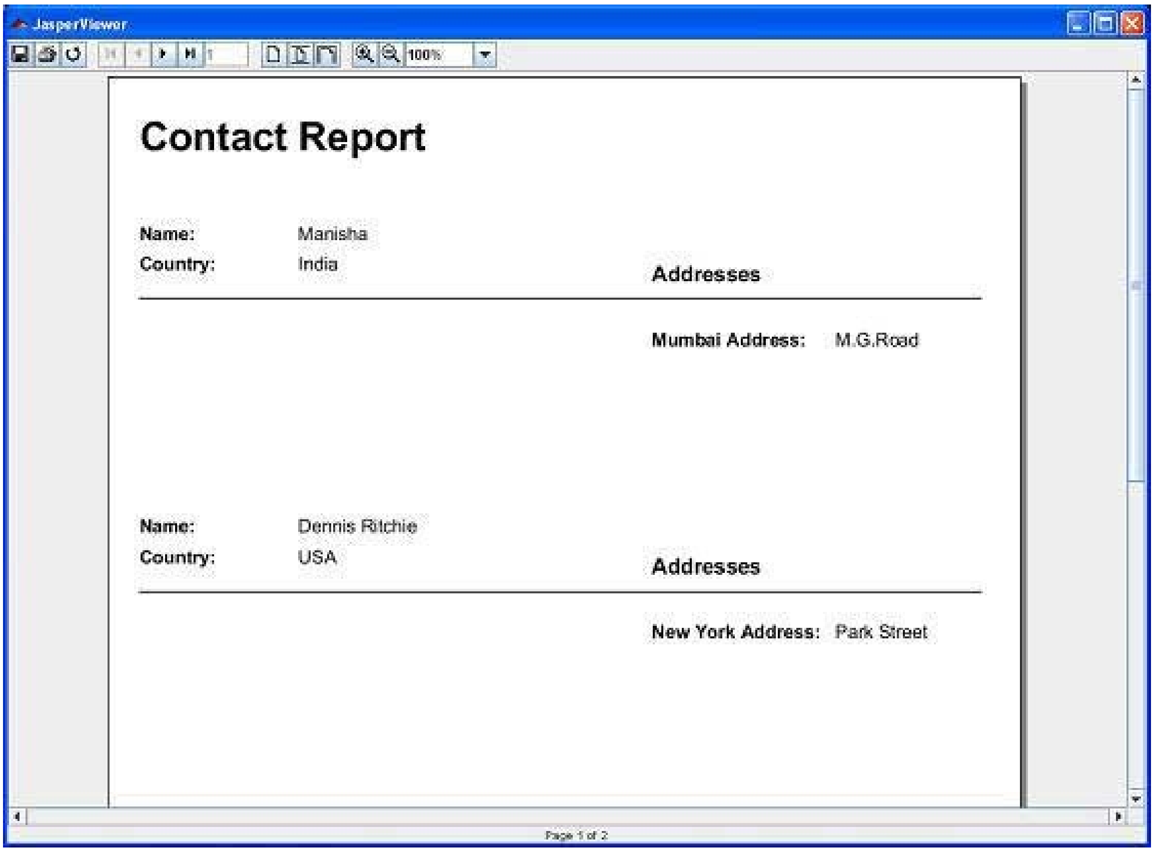
> *The import file - baseBuild.xml is picked from chapter [Environment Setup] and should be placed in the same directory as the build.xml.*

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="JasperReportTest" default="viewFillReport" basedir=".">
   <import file="baseBuild.xml" />
   <target name="viewFillReport"
      depends="compile,compilereportdesing,run"
      description="Launches the report viewer to preview
      the report stored in the .JRprint file.">
      <java classname="net.sf.jasperreports.view.JasperViewer"
      fork="true">
         <arg value="-F${file.name}.JRprint" />
         <classpath ref />
      </java>
   </target>
   <target name="compilereportdesing"
      description="Compiles the JXML file and
      produces the .jasper file.">
      <taskdef name="jrc"
      classname="net.sf.jasperreports.ant.JRAntCompileTask">
         <classpath ref />
      </taskdef>
      <jrc destdir=".">
         <src>
         <fileset dir=".">
            <include name="*.jrxml" />
         </fileset>
         </src>
         <classpath ref />
      </jrc>
   </target>
</project>
```

Next, let's open command line window and go to the directory where build.xml is placed. Finally execute the command **ant -Dmain-class=com.tutorialspoint.CreateReport** (viewFullReport is the default target) as follows:

```
Buildfile: C:\tools\jasperreports-5.0.1\test\build.xml

clean-sample:
   [delete] Deleting directory C:\tools\jasperreports-5.0.1\test\classes

compile:
    [mkdir] Created dir: C:\tools\jasperreports-5.0.1\test\classes
    [javac] C:\tools\jasperreports-5.0.1\test\baseBuild.xml:28: warning: 'includeantruntime'
was not set, defaulting to
    [javac] Compiling 7 source files to C:\tools\jasperreports-5.0.1\test\classes

compilereportdesing:
      [jrc] Compiling 1 report design files.
      [jrc] log4j:WARN No appenders could be found for logger
(net.sf.jasperreports.engine.xml.JRXmlDigesterFactory).
      [jrc] log4j:WARN Please initialize the log4j system properly.
      [jrc] log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
      [jrc] File : C:\tools\jasperreports-5.0.1\test\jasper_report_template.jrxml ... OK.

run:
     [echo] Runnin class : com.tutorialspoint.CreateReport
     [java] Compiling Report Design ...
     [java] log4j:WARN No appenders could be found for logger
(net.sf.jasperreports.engine.xml.JRXmlDigesterFactory).
     [java] log4j:WARN Please initialize the log4j system properly.
     [java] Done filling!!! ...

viewFillReport:
     [java] log4j:WARN No appenders could be found for logger
(net.sf.jasperreports.extensions.ExtensionsEnvironment).
     [java] log4j:WARN Please initialize the log4j system properly.

BUILD SUCCESSFUL
Total time: 72 minutes 13 seconds
```

As a result of above compilation, a JasperViewer window opens up as in the screen below:



Contact Report

Name: Manisha
Country: India

Addresses

Mumbai Address: M.G.Road

Name: Dennis Ritchie
Country: USA

Addresses

New York Address: Park Street

Here we can see that the Name, country and address are displayed.